

软件开发：数组结构

COMPUTING: Software Development

HIGHER NATIONAL DIPLOMA

【英】苏格兰学历管理委员会 (SQA)
Scottish Qualifications Authority

Unit Student Guide

Software Development: Array Data Structures DG8T 04



中国时代经济出版社

SCOTTISH
QUALIFICATIONS
AUTHORITY



软件开发：数组结构

COMPUTING: Software Development

HIGHER NATIONAL DIPLOMA


【英】苏格兰学历管理委员会 (SQA)

Scottish Qualifications Authority

Unit Student Guide

Software Development: Array Data Structures DG8T 04



 中国时代经济出版社

SCOTTISH
QUALIFICATIONS
AUTHORITY



著作权合同登记 图字：01-2004-3453号

图书在版编目 (CIP) 数据

软件开发：数组结构/苏格兰学历管理委员会著.-北京：中国时代经济出版社，2005.2

ISBN 7-80169-641-7

I .软… II .苏… III .软件开发-教材-英文 IV .TP311.52

中国版本图书馆CIP数据核字 (2004) 第135112号

“First published by CMEPH”

“All Rights Reserved”

“Authorized Apograph/ Translation/Adaptation of the editions by the Scottish Qualifications Authority. All Intellectual Property Rights vest in the Scottish Qualifications Authority and no part of these “Works” may be reproduced in any form without the express written permission of Scottish Qualifications Authority”

Software Development: Array Data Structures

软件开发：数组结构

苏格兰学历管理委员会著

出 版 者	中国时代经济出版社
地 址	北京市东城区东四十条24号 青蓝大厦东办公区11层
邮 政 编 码	100007
电 话	(010) 68320825 68320517
传 真	(010) 68320634
发 行	各地新华书店
印 刷	北京新丰印刷厂
开 本	787×1092 1/16
版 次	2005年2月第1版
印 次	2005年2月第1次印刷
印 张	10.75
定 价	28.00元
书 号	ISBN 7-80169-641-7/G·204

版权所有 侵权必究

Contents

1	Introduction to the Scottish Qualifications Authority	1
2	Introduction to the Unit	3
2.1	What is the Purpose of this Unit?	3
2.2	What are the Outcomes of this Unit?	3
2.3	What do I Need to be Able to do in Order to Achieve this Unit?	4
2.4	Approximate Study Time for This Unit	6
2.5	Equipment/Material Required for this Unit	6
2.6	Symbols Used in this Unit	7
3	Assessment Information for this Unit	9
3.1	What Do I Have to Do to Achieve This Unit?	9
4	Suggested Lesson Plan	11
5	Learning Materials	13
5.1	Section 1	13
5.2	Section 2	25
5.3	Section 3	64

6	Additional Reading Material	105
7	Solutions to Self Assessed Questions and Activities	107
8	Supplement for Software Development: Array Data Structures	109
8.1	Section S1 — Introduction	109
8.2	Section S2 — Mappings	113
8.3	Section S3 — Arrays in C	118
8.4	Section S4 — Ordered	126
8.5	Section S5 — Queues	131
8.6	Section S6 — Stacks	136
8.7	Section S7 — Search	146
8.8	Section S8 — Sorting	152
8.9	Section S9 — Other Resources	160
9	Copyright References	163
10	Acknowledgements	165

1

Introduction to the Scottish Qualifications Authority

This Unit DG8T 04 Software Development: Array Data Structures has been devised and developed by the Scottish Qualifications Authority (SQA). Here is an explanation of the SQA and its work:

The SQA is the national body in Scotland responsible for the development, accreditation, assessment, and certification of qualifications other than degrees.

Its website can be viewed on: www.sqa.org.uk.

SQA's functions are to:

- devise, develop and validate qualifications, and keep them under review
- accredit qualifications
- approve education and training establishments as being suitable for entering people for these qualifications
- arrange for, assist in, and carry out, the assessment

of people taking SQA qualifications

- quality assure education and training establishments which offer SQA qualifications
- issue certificates to candidates.

In order to pass SQA units, students must complete prescribed assessments. These assessments must meet certain standards.

The Unit Specification outlines the three Outcomes that students must complete in order to achieve this unit. The Specification also details the knowledge and/or skills required to achieve the outcome or outcomes. The Evidence Requirements prescribe the type, standard and amount of evidence required for each outcome or outcomes.

2

Introduction to the Unit

2.1
What is the
Purpose of
this Unit?

This unit is designed to enable students to become familiar with abstract data types and the array data structures that can be used to implement them within software systems. Two important topics, sorting and searching, are introduced in the context of array based data structures.

2.2
What are the
Outcomes of
this Unit?

There are three outcomes for this unit.

- Describe data representation and storage in computer systems
- Develop and implement operations on array data structures
- Develop, implement and use searching and sorting techniques.

2.3

What do I
Need to be
Able to do in
Order to
Achieve this
Unit?

Achievement of the first outcome of the unit requires the candidate to correctly answer at least 60% of the questions in a multiple choice test. The test will comprise 20 questions and will cover areas such as

- Representation of simple and structured data types
- Representation of complex data types stored externally
- File storage issues relating to data representation.

To achieve this task students will need to understand how data is structured, and may have to investigate how different file structures are implemented.

Outcome 2 requires the candidate to develop and provide fully documented and working code for two different abstract data structures. The abstract data structure to be developed will be selected from:

- Stack
- Queue
- Circular Queue
- Ordered Listed.

In keeping with the theme of abstract data structures, each of the implementations must be implemented using a modular approach, with the operations on the structure implemented as subprograms. Each structure will require at least the following operations.

- Initialise
- Add
- Delete
- IsEmpty
- IsFull.

In order to be able to perform these tasks students will need to be able to design and implement computer programs using a number of language features, especially parameter passing.

Outcome 3 requires the candidate to develop, implement and use searching and sorting techniques. This will involve the candidate in providing one desktop walkthrough and one practical implementation of searching/sorting algorithms. The second part of this will again involve design and implementation of computer programs.

For outcomes two and three some prior experience of a programming language is assumed. In particular the use of arrays, loops, conditional statements and passing parameters into sub-programs will be used extensively. It is assumed that candidates understand the difference between passing arguments by reference and passing arguments by value. Code examples will be given in the C programming language, but most of the pseudocode will be implementation language independent.

2.4 Approximate Study Time for This Unit

It is anticipated that you will meet as a class for approximately 40 hours for this unit, spending this time according to the following schedule :

- Outcome 1: 8 – 10 hours
- Outcome 2: 8 – 10 hours
- Outcome 3: 18 – 20 hours.

It is understood that you will spend approximately 40 hours of individual study time in addition to the time noted above.

It must be clearly understood that developing software is a time consuming activity, with a wide scope for frustrating delays.

2.5 Equipment/ Material Required for this Unit

To complete this unit students will need to be able to access a computer with a programming language translator. The material presented in this unit should be suitable for use with a wide range of imperative programming languages, but will assume the use of the 'C' programming language when code samples are provided. Even within this programming language there exist variations and care must always be taken.

2.6 Symbols Used in this Unit

The various Learning Materials sections are designed so that you can work at your own pace, with tutor support. As you work through the Learning Materials (see Section 5), you will encounter symbols. These symbols indicate that you are expected to do a task. **These tasks are not Outcome Assessments.** They are exercises designed to consolidate learning or encourage thought, in preparation for the Outcome Assessment (see Section 3 — Assessment Information for this Unit).

Activity



This symbol indicates an Activity (A). Usually, activities are used to improve or consolidate your understanding of the subject in general or a particular feature of it.

The activities will not serve this purpose if you refer to the responses prior to having attempted the Activity.

Self Assessed Question



This symbol indicates a Self Assessed Question. Using a Self Assessed Question helps you check your understanding of the content that you have already covered. Everything is provided for you to check your

own responses. Answers to the Self Assessed Questions are to be found at the back of the Unit Student Guide. Where suggested responses to activities are provided in the Unit Student Guide, **students are strongly discouraged from looking at these responses before they attempt the activity.** The activities throughout the Unit Student Guide will help you to prepare yourself for the formal assessments, and to identify topic areas in which you will require clarification and additional tutor support. The activities will not serve this purpose if you look at the answers before trying the activity!

Self Assessed Questions and activities are designed to be checked by you. No tutor input is necessary at this stage unless special help is requested, although from time to time your tutor may wish to view your responses to Self Assessed Questions to see how you are progressing.

3

Assessment Information for this Unit

3.1
What Do I
Have to Do to
Achieve This
Unit?

Outcome 1

You will be presented with a multiple choice question paper with twenty questions about data representations. To achieve the outcome you will have to answer at least 60% of these questions correctly. This will be conducted under examination conditions—you will not be required to refer to books, or other sources.

Outcome 2

You will be required to design, document and provide array based implementations of two different abstract data structures. This is a practical based assessment and you may refer to books and other sources. You should be prepared to discuss your submission with an assessor to allow the assessor to confirm that the work is authentic.

Outcome 3

You will have to provide a desk checked version of one

sorting technique and one search technique. You will conduct the desk checks in the presence of an assessor.

You will also have to provide an implementation of a different sorting technique and an implementation of a different search technique. This is a practical based assessment and you may refer to books and other sources. You should be prepared to discuss your submission with an assessor to allow the assessor to confirm that the work is authentic.

4

Suggested Lesson Plan

The Learning Materials (see Section 5) are designed to lead you through a series of activities which will allow you to consolidate your learning and check on your own progress.

It is suggested that the material in this guide is used in sequence, with the outcomes being studied in order. The timing suggested here are advisory only, in particular some centres may find that Outcome 1 requires less than the suggested time, and may want to make more time available for the implementation sessions for outcomes 2 and 3.

Outcome 1

- Hour 1 — Introduction to and discussion of data types
- Hour 2 — Introduction to and discussion of storage size issues
- Hour 3 — Introduction to and discussion of mapping functions and arrays
- Hour 4 — Introduction to and discussion of structures
- Hour 5 — Introduction to and discussion of data

structures in files

Hour 6 — Revision

Hour 7 — Assessment

Hour 8 — Review/Remediation.

Outcome 2

Hour 1 — Introduction to Abstract data structures

Hour 2 — Stack

Hour 3 — Queue

Hour 4 — Circular Queue

Hour 5 — Ordered List

Remaining time— implementation and assessment.

Outcome 3

Hour 1 — Introduction and discussion of searching

Hour 2 — Sequential and binary search.

Hour 3 — Introduction and discussion of sorting

Hour 4 — BubbleSort and InsertionSort

Hour 5 — SelectionSort and ShellSort

Hour 6 — HeapSort and MergeSort

Hour 7 — Revision

Hour 8 — Walkthrough/deskcheck assessment

Remaining time— implementation and assessment.